SLP's MLP in NLP

Toxic Comment Classification Challenge - A Kaggle Competition

Presentation by

Lakshmi Prabha S.,

NYCDSA boot camp student.

Overview

- The Conversation AI team, a research initiative founded by Jigsaw and Google (both a part of Alphabet) are working on tools to help improve online conversation.
- “Discussing things you care about can be difficult. The threat of abuse and harassment online means that many people stop expressing themselves and give up on seeking different opinions.”

- **Platforms struggle to effectively facilitate conversations, leading many communities to limit or completely shut down user comments”.**

- **One area of focus is the study of negative online behaviors, like toxic comments (i.e. comments that are rude, disrespectful or otherwise likely to make someone leave a discussion).**

Literature Review

- **So far they've built a range of publicly available models served through the Perspective API, including toxicity.**
- **But the current models still make errors, and they don't allow users to select which types of toxicity they're interested in finding (e.g. some platforms may be fine with profanity, but not with other types of toxic content).**

- **In this competition, the challenge is to build a multi-headed model that's capable of detecting different types of toxicity like threats, obscenity, insults, and identity-based hate.**

Data Description

- A large number of Wikipedia comments are provided which have been labeled by human raters for toxic behavior. The types of toxicity are:
  - toxic
  - severe_toxic
  - obscene
  - threat
  - insult
  - identity_hate
- Aim: To create a model which predicts a probability of each type of toxicity for each comment.

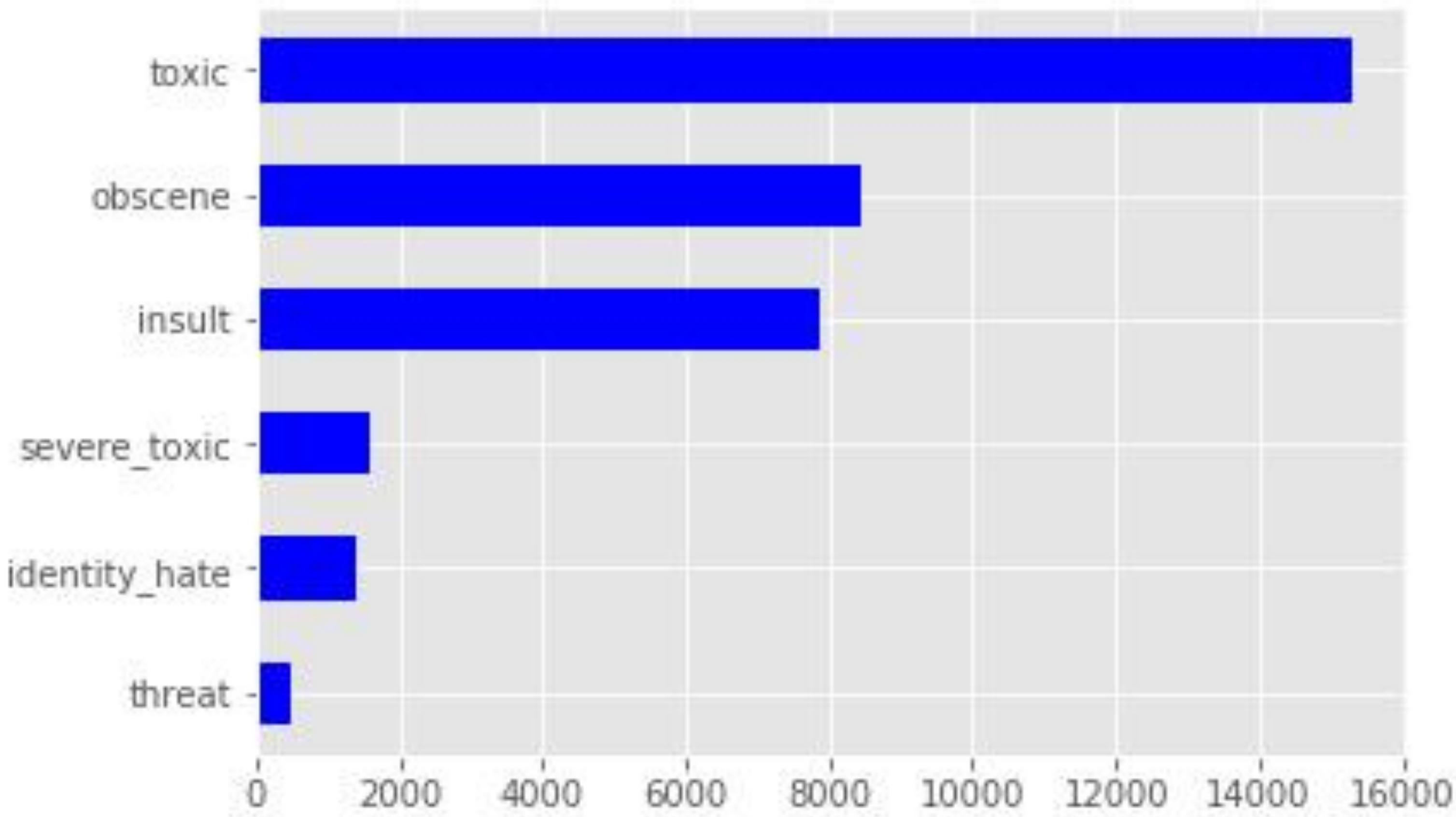
Evaluation Metric

- **Update: Jan 30, 2018. Due to changes in the competition dataset, we have changed the evaluation metric of this competition.**
- **Submissions are now evaluated on the mean column-wise ROC AUC. In other words, the score is the average of the individual AUCs of each predicted column.**

My Question:

- **What is the potential difference in toxicity (or conversation heat) between the classes?**

- **Distribution of the classes:**



Models - I
One Vs Rest Classifier

- **Train-test-split : 70% - 30%**
- **<u>Pipeline:</u>**
  - **Count Vectorizer or Tfidf Vectorizer**
  - **GridSearchCV**
  - **Classifier**
    1. **Logistic Regression: Accuracy: 0.918**
    2. **Random Forest Classifier: 0.915**
    3. **Gradient Boost Classifier: 0.908**
    4. **MultinomialNB: 0.905**
    5. **SVC: 0.896**

Models - II

Binary Classifier for each Label

Feature Engineering

- **Tfidf Vectorizer**
  - **Part (i): analyzer = ‘word’**
  - **Part (ii): analyzer = ‘char’ (to deal with foreign languages)**
- **Stacking (combining): word features + char features**
- **Hyper-parameters:**
  - **ngram range – (1,1)**
  - **min_df = 0.0001**
- **Max: 62311 + 2500 features**

Model Selection

- Logistic Regression: 0.9752 (Initial Kaggle submission)
- SVC: 0.923
- MultinomialNB: 0.88
- Random Forest Classifier: 0.855
- Gradient Boost Classifier: 0.85

Preprocessing + Feature Engineering + Hyper parameter Tuning

Patterns for each class:

- **Threat:** **‘kill’, ‘shoot’, ‘murder’, ‘gun’, …**
- **Identity-hate:** **‘Nigerian, Jews, Muslim, gay’,…**
- **Obscene:** **vulgar or offensive words**
- **Severe Toxic:** **Offensive + Hurtful**
- **Insult:** **Noun or Pronoun + vulgar or offensive or hurtful words**
- **Toxic:** **All others,**

  ‘sorry’, ‘thanks’ or any general discussion

- Removal of stop words (English)
- Removal of Punctuation
- Important words (features) selection for each class using Tfidf Vectorizer (diff. min_df).
- Keeping only important words in the text – 4 hours
- Analyzer = ‘word’:
- ngram range (1,1): 25265 features
- Analyzer – ‘char’:
- ngram range (5,6): 28000 features

- Logistic Regression - Total Accuracy: 0.977
- MultinomialNB – Total Accuracy: 0.908 (from 0.88)
- Aim: 0.9792

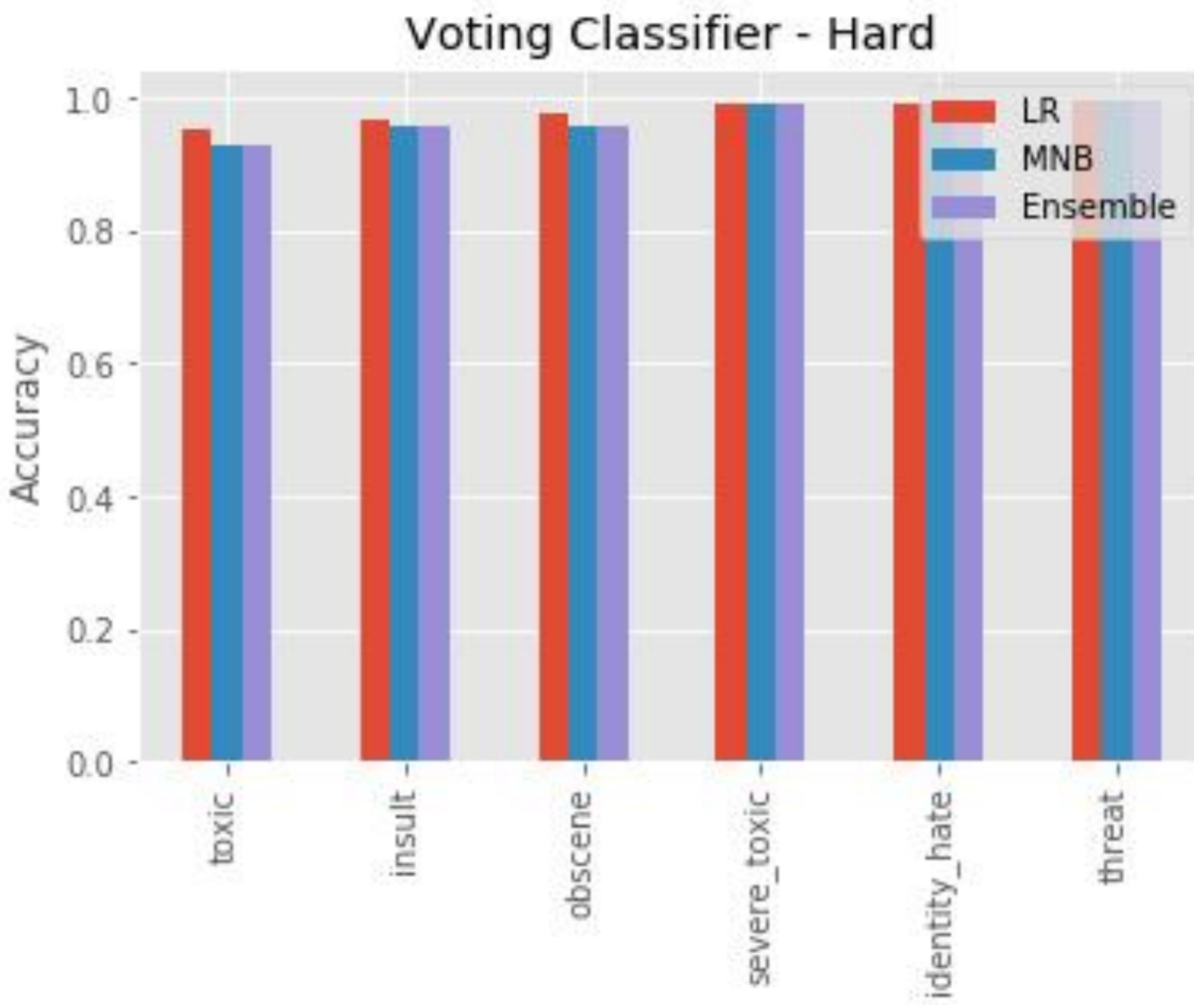
Vitaly Kuznetsov NIPS2014

https://mlwave.com/kaggle-ensembling-guide/

Ensembling – Voting Classifier - Soft

- **Weighted Average (without re-training model)**
  - **My First Submission: 0.9752**
  - **Important Words Selection: 0.977**
  - **From Kaggle Kernels: 0.97929877**
- **Weights:**
  - **c = 100**
  - **d = 1000**
  - **e= 300000000000000000**
- **Accuracy: 0.97921 (Kaggle Submission 2)**

Ensembling –
Voting Classifier - Hard



Model - III

Stacking

- Import submission file from Kaggle Kernels: 0.9854
- Set threshold = 0.5
- Re-train the model with the entire data set and predict
- Search for the threshold and choose the best threshold for each class such that the difference in predicted probabilities between my model and imported model is the minimum.

- **Accuracy: 0.9784 & 0.9788**

  (Kaggle Submissions 3 & 4)

Kaggle Submission 5

- **Weighted Average (without re-training model)**
  - **My Submission 4: 0.9788**
  - **My Submission 2: 0.9792**
  - **From Kaggle Kernels: 0.9854**
- **Weights:**
  - **b = 0.3**
  - **c = 0.3**
  - **d = 0.4**
- **Accuracy: 0.9843**

Kaggle Submission 6

- **Weighted Average (without re-training model)**
  - **My Submission 5: 0.9843**
  - **From Kaggle Kernels: 0.9854**
- **Weights:**
  - **b = 0.5**
  - **c = 0.5**
- **Accuracy: 0.9851**

Final Kaggle Submission (for this project)

- **Repeating the above process…**
- **Weighted Average (without re-training model)**
  - **My Submission : 0.9851…**
  - **From Kaggle Kernels: 0.9854**
- **Weights:**
  - **b = 0.5**
  - **c = 0.5**
- **Accuracy: 0.9854**

Observations:

- Classes ‘Obscene’ and ‘Threat’ are easy to classify.
- Deeper analysis are required for finding patterns in ‘Toxic’ and ‘Insult’ classes.
- Standard ML algorithms yielded a maximum of 0.9792.
- To improve the accuracy further, one has to use DL techniques.
- Simple DL technique yielded just 0.977.
- DL + various ensemble techniques yielded 0.98…

Future Directions:

- **Apply Deep Learning and model the problem using**
  - **RNN networks**
  - **CNN neworks**
  - **Bi-directional RNN + GRU**
  - **FastText (of face book) and Glove (of Google)**
- **Analyze the difference between the classes in depth.**

References:

- **https://mlwave.com/kaggle-ensembling-guide/**
- **https://www.kaggle.com/tunguz/logistic-regression-with-words-and-char-n-grams**
- **https://www.kaggle.com/tunguz/blend-of-blends-1/output**

THANK YOU!!!

SLP’s MLP in NLP